

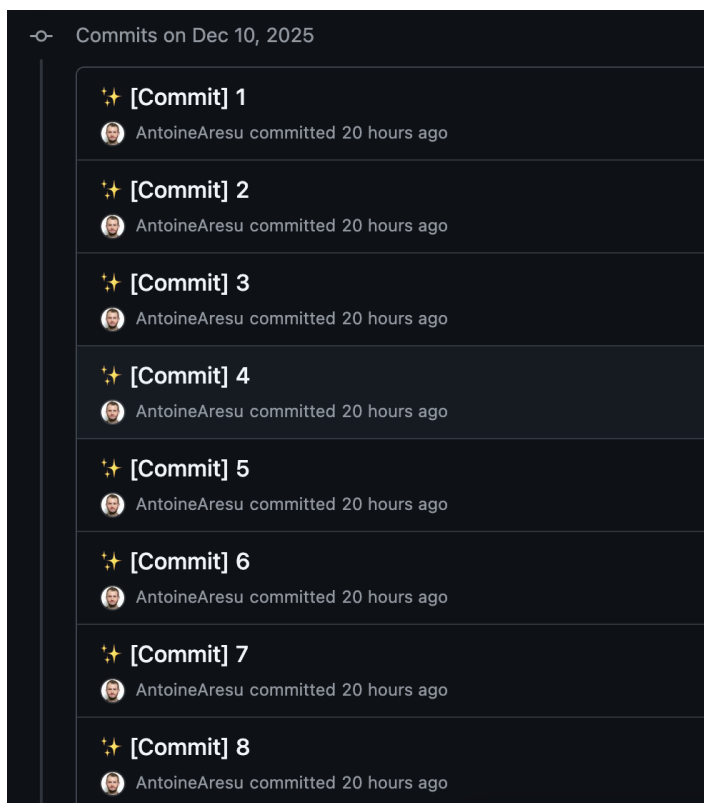
git bisect

À quoi ça sert ?

- Permet de mettre en évidence un commit qui a induit un changement dans un historique de commits
- Ce changement peut correspondre à un bug, à un test qui casse... tout changement d'état

Comment ça marche

- Prenons l'exemple le plus simple : j'ai développé une feature, et j'ai induit un bug
- Je sais qu'au premier commit (commit 1), je n'avais pas de bug
- Je sais qu'au dernier commit (commit 8), j'ai un bug



Au lieu de checkout sur chaque commit pour constater si le bug existe (ce qui impliquerait de vérifier 8 commits), je peux utiliser git bisect :

- Je vais devoir flaguer un commit "good" (sans bug) -> **commit 1**
- Et un commit "bad" (avec bug) -> **commit 8**

- Ensuite, au lieu de vérifier commit par commit, git bisect va séparer la feature en 2 : il va checkout sur le commit 4 pour savoir si il est "good" ou "bad".
- Et ainsi de suite ! Si le commit 4 est "good", le bug se trouve forcément entre le commit 4 et le commit 8, donc il va me faire vérifier le commit 6 etc... Jusqu'à pouvoir identifier le commit qui a induit une erreur

En pratique :

Entrer en mode bisection :

```
git bisect start
```

Flag initial d'un commit "good" et "bad", en indiquant la référence du commit (optionnel si je me trouve sur ce commit :

```
git bisect good cebcb8b352b80af4dfa92e554ac5a608dffa7be1 #(SHA commit 1)
git bisect bad #le SHA peut être optionnel si je suis positionné sur le commit 8
```

Flag des commits proposés durant la bisection :

```
git bisect good
#OU
git bisect bad
```

Fin de la session bisection :

```
git bisect reset
```

Ceci correspond au mode manuel de git bisect, mais il est aussi possible de l'automatiser. Pour cela, reproduire les 2 premières étapes mentionnées plus haut. Puis lancer :

```
git bisect run <monScript>
```

Le script peut être n'importe quoi : lancer un test unitaire, exécuter un script php, bash etc... La bissection sera alors automatique, et git bisect pourra flaguer les commits automatiquement en fonction du retour du script, et donc mettre en évidence le commit problématique seul

Révision #3

Créé 11 décembre 2025 10:36:53

Mis à jour 6 mai 2026 08:41:24